

文章编号: 1001- 2486(2008) 02- 0091- 06

双正交重叠变换系数基于上下文的算术编码算法*

罗武胜, 王继东

(国防科技大学 机电工程与自动化学院, 湖南 长沙 410073)

摘要: 提出了一种双正交重叠变换(LBT)系数基于上下文的算术编码算法。针对二进整数 LBT 在 DSP 实现过程中出现的精度和计算复杂度问题, 提出了一种适于 DSP 并行处理的 LBT 定点实现方法。编码算法包含 60 种上下文概率模型, 熵编码采用 MQ 编码, 并根据并行处理的需要, 将 MQ 编码从位平面扫描过程中分离, 根据分离后的结构特点, 设计了改进的 MQ 编码器。并对算法的 DSP 实现进行了研究。实验结果表明, 本压缩算法压缩性能和 SPIHT 相当, 稍逊于 JPEG2000, 算法容易并行, 硬件计算复杂度很低。

关键词: 双正交重叠变换; 上下文概率模型; 算术编码; 并行处理

中图分类号: TN919. 8 **文献标识码:** A

Context Based Arithmetic Coding of Lapped Transform
Biorthogonal Coefficients

LUO Wusheng, WANG Jidong

(College of Mechatronics Engineering and Automation, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: A context based arithmetic coding of Lapped Biorthogonal Transform (LBT) coefficients is proposed. Aiming at the shortcut of precision and computation complexity of binary integer LBT in DSP implementation, a fixed point algorithm of LBT which fits parallel processing in DSP is proposed. There are 60 context models in the algorithm. The coder of entropy coding is MQ-coder. For the purpose of parallel processing, the MQ-coding is isolated from the bit plane scanning. Based on the characteristic of coding algorithm after isolated coding, a modified MQ-coder is designed. Meanwhile, the implementation in TMS320C6000 DSP of the algorithm is studied. Experiments show that the algorithm's coding performance is mostly the same as SPIHT and very competitive with JPEG2000. The tests on the hardware platform show that the algorithm is simple for parallel processing and the computation complexity is very low.

Key words: lapped biorthogonal transform; context probability modeling; arithmetic coding; parallel processing

针对 DCT 用于图像压缩时出现的块效应问题, 重叠技术被引入图像压缩技术中, 其原理是实现信号的部分重叠处理。典型的两类重叠变换都是基于 DCT, 一类是在 DCT 变换后的频域进行重叠变换^[1], 一类是在 DCT 前直接在时域进行重叠变换^[2], 常称为后处理和预处理, 两类变换在国内被习惯地叫做双正交重叠变换(LBT)。

为了进一步降低复杂度, 对于 DCT 和 LBT, 人们都利用提升方法发展了它们的二进整数实现^[3-4], 效果和浮点接近。现有的 LBT 系数编码方法大部分由 DCT 系数的编码方法和离散小波变换(DWT)系数的编码方法继承而来。嵌入式零树编码和基于上下文的算术编码被引入到基于 DCT 和 LBT 的图像压缩中^[5-6], 取得与基于 DWT 的压缩方法相似的效果。

针对二进整数 LBT 的不足, 提出了 LBT 的定点实现方法, 复杂度降低且近似更好。并根据 LBT 的系数特性, 发展了基于上下文的算术编码算法, 其压缩效果和 JPEG2000 相近, 且计算复杂度较低。

1 双正交重叠变换的定点实现

进行重叠变换时, 输入向量的长度大于输出向量的长度, 相邻块间存在重叠。从滤波器分解的角度

* 收稿日期: 2007- 09- 24

基金项目: 国家 863 高技术基金资助项目(2006AA 701121)

作者简介: 罗武胜(1972—), 男, 副教授, 博士。

来看,LBT利用了相邻系数块间的相关性,能量集中性好,从而可以更有效地对变换系数进行编码。

1.1 LBT的二进制整数实现(binLBT)

整数算法可以进一步降低双正交重叠变换的计算复杂度,并在低压缩比时构造无损压缩。同时,采用整数算法时变换系数占用16位,而浮点算法需要占用64位,算法所需存储量也大大减小。LBT的整数实现以前常用的是二进制整数实现方法(binLBT)。

信息处理中广为采用的各种正交变换从代数角度看,均能分解为平面旋转矩阵的乘积。为了将变换矩阵分解为提升步,首先需要把平面旋转矩阵分解成提升矩阵的乘积,具体分解式为

$$R_0 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \tan\frac{\theta}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sin\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tan\frac{\theta}{2} & 1 \end{bmatrix} \quad (1)$$

则提升矩阵及其逆均能很容易地通过提升步近似,此时只需要移位与二进的整数运算。

在对二进制整数LBT实现的过程中发现:为了最佳逼近,对移位的运算需要增加四舍五入处理,例如: $x \times 1/8 = (x + 4) \gg 3$; 实现无乘法运算时形如 $3/8$ 的分数要进行分解 $3/8 = 1/4 + 1/8$; 提升分解以后的整数变换对浮点的逼近精度不足,造成恢复图像质量的相应差距。

目前高性能的DSP、ARM等处理器均有完备的乘法运算指令,FPGA等也具有乘法操作能力,在硬件平台上的算法实现把乘法分解为多个移位和加法不再必要。由于处理器可以进行16位和32位的操作,浮点截断的有效位可以取得较高,显然大于二进整数的舍入精度。由此,提出了LBT的定点实现方法。

1.2 LBT的定点实现算法

定点算法主要对数值计算截断的过程进行算法上的设计和优化。定点算法核心设计思想是使用定点数代替浮点数,并使用定点运算代替浮点运算。DSP中通常用Q格式数表示小数。小数乘一个比例因子并取整变为整型数后,使用定点运算指令,运算完毕后除以该比例因子。

LBT实现过程中的主要运算为行矩阵和平面旋转矩阵相乘,首先将平面旋转矩阵中的三角函数值左移16位,转化为Q16格式数,运算完毕后右移16位。正弦、余弦函数值的Q16格式数分别表示为 $ic\theta$ 和 $is\theta$: $ic\theta = (\text{int})(\cos\theta \ll 16)$, $is\theta = (\text{int})(\sin\theta \ll 16)$ 。

$$\begin{aligned} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \cos\theta - x_2 \sin\theta \\ x_1 \sin\theta + x_2 \cos\theta \end{bmatrix} \\ &\approx \begin{bmatrix} (x_1(\cos\theta \ll 16) - x_2(\sin\theta \ll 16)) \gg 16 \\ (x_1(\cos\theta \ll 16) + x_2(\sin\theta \ll 16)) \gg 16 \end{bmatrix} = \begin{bmatrix} (x_1 \times ic\theta - x_2 \times is\theta) \gg 16 \\ (x_1 \times is\theta + x_2 \times ic\theta) \gg 16 \end{bmatrix} \end{aligned} \quad (2)$$

时域双正交重叠变换的实现流程如图1所示。

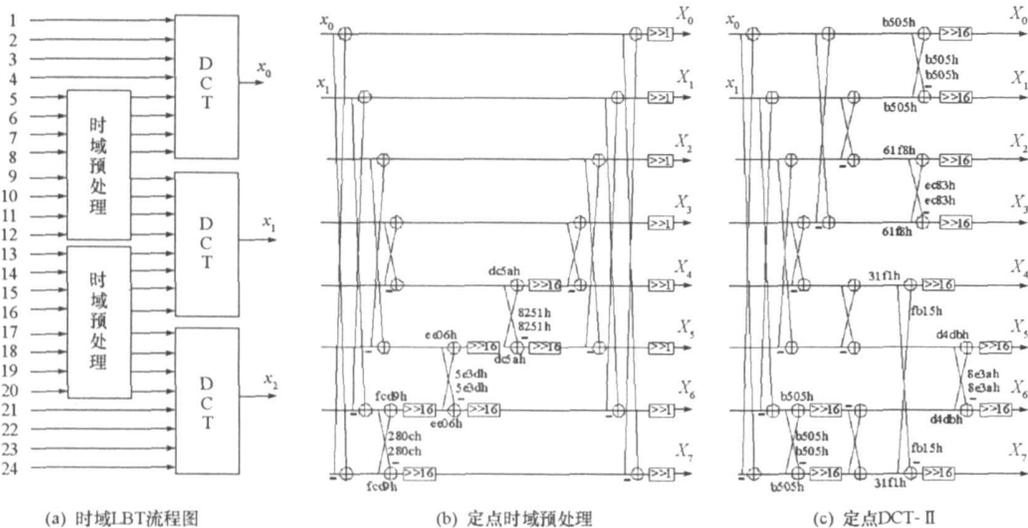


图1 时域双正交重叠变换的定点实现

Fig. 1 Flow diagram of fixed point LBT in time domain

2 块变换系数中基于上下文的熵编码算法

2.1 块变换系数基于上下文的熵编码算法(CEB)

文献[6]提出了一种块变换系数基于上下文的熵编码算法(CEB, Context-based Entropy Coding of Block Transform Coefficients)。文献[6]中,系数块的子带被分成七类,如图2所示。

算术编码按从最高比特平面到最低平面的顺序进行,直到达到给定的压缩率或失真程度。在每个比特平面,各个块按从左到右从上到下的顺序。对于第 b 个比特平面,每个块的编码方法进行细化编码、重要性编码、符号编码和集合编码。CEB算法充分考虑了块内、块间系数的关系和符号关系,以此建立上下文模型,采用自适应二进制算术编码,取得了良好的图像压缩效果。

CEB算法包含L-CEB和E-CEB两种编码算法,分别有42种和118种上下文模型。E-CEB上下文模型简单,并采用标量量化,L-CEB算法计算复杂度较低,但这也造成重建图像的差距。E-CEB算法计算复杂度过高,不适合一些需要实时压缩的场合。CEB算法以Q编码器为基础,不是符号的非二进制系数进行二进位化。新的静止图像压缩标准JPEG2000采用MQ编码,可以考虑采用MQ编码器代替Q编码器。

本文借鉴了CEB算法基于上下文的熵编码思想,用于LBT系数的编码。并针对CEB算法存在的不足,做出一定的改进,提出LBT系数基于上下文的算术编码算法。

2.2 上下文概率模型的确定

上下文模型的建立考虑LBT系数特点,集中于左块上块相邻系数重要性、块内左方上方系数重要性、左块上块相邻系数符号重要性的联系,且系数符号重要性进行了进一步区分,共60种上下文模型。

设 $C_{x,y}^{i,j}$ 为LBT系数, (x,y) 为系数块的位置坐标,其左、上邻块的位置为 $(x-1,y)$ 和 $(x,y-1)$; (i,j) 为系数在块中的坐标,在块中其左邻和上邻系数的位置为 $(i-1,j)$ 和 $(i,j-1)$ 。设 l 为系数 $C_{x,y}^{i,j}$ 的Zigzag索引值, $C_{x,y}^{i,j}$ 也定义为 $C_{x,y}^l$ 。 $S_{x,y}^l$ 为系数的集合: $S_{x,y}^l = \{C_{x,y}^l, C_{x,y}^{l+1}, \dots, C_{x,y}^{63}\} = \{C_{x,y}^l, S_{x,y}^{l+1}\}$ 。算术编码按从最高比特平面到最低平面的顺序进行, b 为当前比特平面标志。定义:

(1) 系数重要性

$$[C_{x,y}^l]^b = \begin{cases} 1, & |C_{x,y}^l| \geq 2^b \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

(2) 邻域系数符号重要性

$$[C_{x,y}^l]^b = \begin{cases} 0, & C_{x,y}^l \geq 2^b \\ 1, & -2^b < C_{x,y}^l < 2^b \\ 2, & \text{otherwise} \end{cases} \quad (4)$$

(3) 系数符号

$$S(C_{x,y}^l) = \begin{cases} 0, & C_{x,y}^l > 0 \\ 1, & C_{x,y}^l < 0 \end{cases} \quad (5)$$

(4) 系数集合重要性

$$[S_{x,y}^l]^b = \begin{cases} 1, & \text{otherwise} \\ 0, & [C_{x,y}^i]^b < 2^b, i \text{ 是集合}, l \leq i \leq 64 \end{cases} \quad (6)$$

分类方法如下:

(1) 系数集合重要性的类别

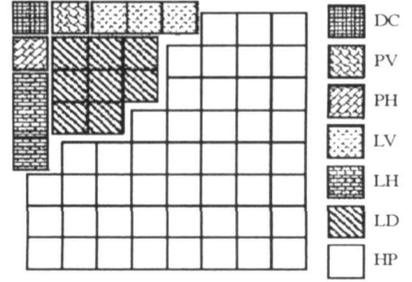


图2 子带划分

Fig. 2 Subband classification

$$\text{type}\left(\left[S_{x,y}^l \right] \right) : \left[S_{x-1,y}^l \right] + \left[S_{x,y-1}^l \right] + \begin{cases} 0, & l = 1 \\ 3, & l = 2 \\ 6, & 2 < l < 15 \\ 9, & l > 14 \end{cases} \quad (7)$$

(2) 系数重要性的类别

$$\text{type}\left(\left[C_{x,y}^{i,j} \right] \right) = \left[C_{x-1,y}^{i,j} \right] + \left[C_{x,y-1}^{i,j} \right] + \begin{cases} 12, & C_{x,y}^{i,j} \in DC \\ 15, & C_{x,y}^{i,j} \in PV \\ 18, & C_{x,y}^{i,j} \in PH \\ 21 + \left[C_{x,y}^{i-1,j} \right], & C_{x,y}^{i,j} \in LV \\ 25 + \left[C_{x,y}^{i,j-1} \right], & C_{x,y}^{i,j} \in IH \\ 29 + \left[C_{x,y}^{i-1,j} \right] + \left[C_{x,y}^{i,j-1} \right], & C_{x,y}^{i,j} \in LD \\ 34 + \left[C_{x,y}^{i-1,j} \right] + \left[C_{x,y}^{i,j-1} \right], & C_{x,y}^{i,j} \in HP \end{cases} \quad (8)$$

(3) 系数符号的类别

$$\text{type}\left(S\left(C_{x,y}^l\right)\right) : \left(C_{x-1,y}^l\right) + \left(C_{x,y-1}^l\right) + \begin{cases} 39, & C_{x,y}^l \in DC \\ 44, & C_{x,y}^l \in PV \cup LV \\ 49, & C_{x,y}^l \in PH \cup IH \\ 54, & C_{x,y}^l \in LD \cup HP \end{cases} \quad (9)$$

2.3 基于并行处理要求的编码分离

位平面编码时,从最高有效平面开始,依次对每个位平面的系数位进行算术编码,由此得到嵌入式的压缩码流。编码过程中对每一位编码包括三个步骤:(1)确定它的上下文;(2)估计上下文的概率;(3)将位和它的概率传送给算术编码器。简化的位平面编码流程如图3(a)所示。对于第(3)个步骤MQ编码而言,各个位和位的概率的编码是相互独立的,可以并行处理。因此,将第(3)个步骤分离出来,第(2)步只要按照第(1)步确定的上下文对该模型下的0或1进行计数,扫描完成后各上下文的概率可以根据记数的结果得到。分离后的位平面编码流程如图3(b)所示。

2.4 改进的MQ编码器

由于编码分离以后概率估计不同,结合硬件实现的特点,对MQ编码器作如下处理或改进:

2.4.1 概率估计

在位平面扫描后独立的编码过程中,不需要进行信源符号的互换,始终约定MPS为1,LPS为0。也不需要查表方式得到 Q_e 的值,在位平面的扫描的过程中,记录了每种上下文模型的二进制数据0,1的个数。扫描完成后,通过归一化可以准确得到各个上下文模型中0,1的概率。即 $p(0|s) = n(0,s)/n(s)$; $p(1|s) = n(1,s)/n(s)$;其中 $n(0,s)$ 和 $n(1,s)$ 分别表示二进制输入数据流 s 中0和1的个数,而 $n(s) = n(0,s) + n(1,s)$ 。如果是定点处理器,可以将概率通过 Q 格式数的方式选择统一的比例因子进行定标,转化为整数进行运算。

2.4.2 有限精度

为了充分利用硬件资源和提高运算速度,考虑减少码字进位处理的次数,让 B 寄存器每次输出数

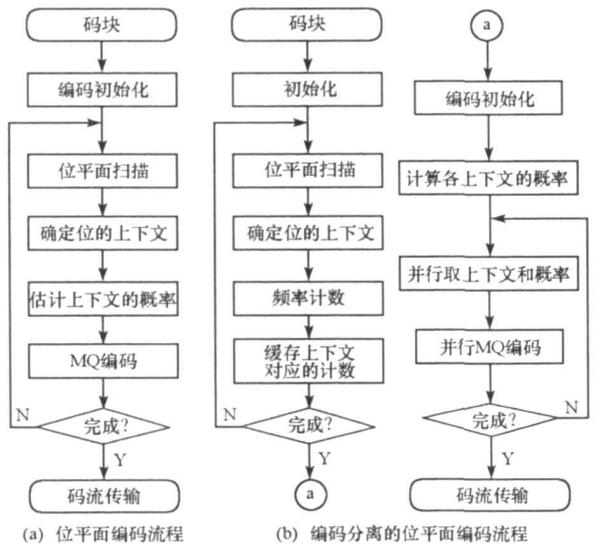


图3 编码分离前后的位平面编码流程

Fig. 3 Bit plane coding flow of pre- and post-isolate coding

据的位数由 8 位改为 16 位, 利用 C 寄存器的全部 32 位, 有限精度处理如图 4 所示。对应的 0x4000 代表小数 0.75, 初始值做相应的变动: $A = 0x4000$ 、 $C = 0$ 、移位计数器 $CT = 15$ 。

2.4.3 码字进位传递的处理

从寄存器 C 中移出的 16 位, 在送入数据流之前, 保存在寄存器 B 中。当 CT 记数减到 0 的时候, 存储在寄存器 B 中的数据输出到压缩数据流中, 然后从寄存器 C 的高位移出 16 位到 B, 如图 4 所示。x 代表 C 中因为 Q_e 的加入而增加的小数位, b 是要从寄存器 C 转移到寄存器 B 的数据位, c 代表进位位。对寄存器 B 的操作遵循位填充机制:

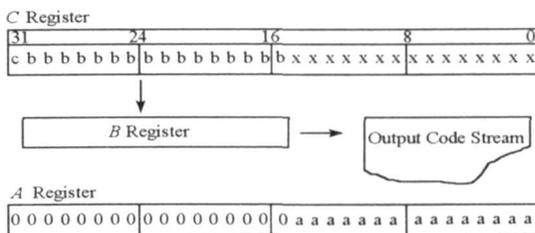


图 4 改进 MQ 编码中的寄存器和输出流
Fig. 4 Register and output of modified MQ-coder

(1) 如果 $B \neq 0xffff$, 并且进位位 $c = 0$, 则把寄存器 B 中的数据位转移到输出数据流, 把寄存器 C 中的第 15 到 30 位的数据转移到寄存器 B, 置 $CT = 16$ 使下一次的输出也是 16 位数据。

(2) 如果 $B = 0xffff$, 则把寄存器 B 中的数据位转移到输出数据流, 把寄存器 C 中包括进位的第 16 到 31 位的数据转移到寄存器 B, 置 $CT = 15$ 使下一次输出包含本次留下的第 15 位。

(3) 如果 $B \neq 0xffff$, 并且进位位 $c = 1$, 则寄存器 B 中的数据加 1, 置进位位 $c = 0$ 。检测寄存器 B, 如果 $B = 0xffff$, 遵循规则 (1), 否则, 遵循规则 (2)。

按照上面的位填充机制, 在解码的时候, 解码器检测所有的数据为 0xffff 的字节后面的第一位, 如果这一位是 1, 则解码器知道这里产生了一个进位。

2.5 算法在星载遥感图像数据压缩机中的应用

本文算法已应用于某卫星图像数据压缩机中。该卫星星载多个线阵 CCD 相机, 相机时钟频率为几十 MHz, 每帧数据为 $n \times k$ 字节, 总码速率近 1Gbps, 要求 4:1 和 8:1 压缩。数据压缩机硬件采用 FPGA+ 多 DSP 结构, 由 DSP 完成压缩算法的实现, DSP 选择 TI 公式的 TMS320C6416 型。在 600MB 主频时, C6416 的最大处理能力达 4800MIPS(百万条指令每秒)。该型号 DSP 的最大优势是指令的并行处理, 对任务独立的算法的处理能力极强。因此对本文算法中 LBT 的并行计算和编码分离以后 MQ 编码的并行处理效率很高。采用的并行策略主要有:

(1) 任务并行分配。LBT 计算时每次处理 16 个点, MQ 编码时每次取 4 对位和位概率, 分别安排在 A 和 B 两个数据通路, 两个通路的运算可以完全并行。

(2) 同址运算。同址运算不仅可以减少内存使用, 还可以消除搬移需要的时间。

(3) 数据打包。程序需要对一连串 short 型数据进行存取, 因此使用 32 位 int 型一次访问 2 个 short 型数据, 减少内存访问, 提高存取速度, 这种优化方法称作数据打包处理。由于 C64x 可在每个时钟周期加载两个(A、B 侧) 64 比特数据, 存取时可以一次访问 4 个 short 型数据。

(4) 多功能指令。程序中主要用到的加、减、绝对值、最大值、移位等操作采用 ADD2、SUB2、ABS2、MAX2、SHR2、SHL2 等半字运算指令, 需要的时候进行打包、解包指令操作。

(5) 优化选项。C6000 系列的汇编优化器可以对代码进行自动优化。-o3 选项表示可得到的最大程度的优化, 使用时尽量联合使用 -pm 选项。

3 实验结果和结论

3.1 客观质量评价

选择标准测试图像(512x512) Lena、Barbara 和 Goldhill, 以峰值信噪比(PSNR)为评价标准, 表 1 列出了本文算法与 SPIHT、JPEG2000 的比较。

表1 三种算法的 PSNR 对比(dB)

Tab.1 Comparison of PSNR of 3 coding algorithm(dB)

算法	图片	0.5bps	1bps	2bps	图片	0.5bps	1bps	2bps	图片	0.5bps	1bps	2bps
本文算法		36.42	39.53	43.78		32.52	37.38	43.20		31.81	36.02	41.06
SPIHT	Lena	36.84	39.98	44.35	Barbara	30.94	35.94	42.05	Goldhill	32.71	36.00	41.11
JPEG2000		37.27	40.34	44.76		32.30	37.18	43.14		33.24	36.57	41.92

3.2 定点叠式变换的计算复杂度

表2列出了五种变换在每个像素上的平均运算量。

表2 五种变换在每个像素上的平均运算量对比

Tab.2 Contrast in operation amounts of 5 transform methods

	浮点 9/7 小波	二进 9/7 小波	二进 LBT	浮点 LBT	定点 LBT
整数加数次数	0	34.125	15	0	5
移位次数	0	23.625	6.75	0	1.5
整数乘数次数	0	0	0	0	1.75
浮点加数次数	10.5	0	0	10.5	0
浮点乘数次数	7.875	0	0	4.625	0

3.3 硬件计算复杂度

以 512×512 的 Lena 为标准图像, 比特率 1, 在工程样机硬件平台上进行测试, 基于 TMS320C6416 型 DSP 的硬件 Timer, 表3给出了算法各部分所需要的时钟周期 (clock) 和并行加速比, 加速比为采用 C 语言串行处理和汇编语言并采用各种并行处理手段后时间比例, 主频 600MB 时压缩 Lena 只需 33ms。

表3 算法计算复杂度

Tab.3 The computation complexity of the algorithm

任务	clock	加速比	任务	clock	加速比
行变换	500396	15.82	最大值寻找	1224213	20.60
列变换	501120	25.65	比特平面扫描	14861208	3.40
Zig-zag 索引	408070	16.03	算术编码	2345396	24.45

4 结束语

本压缩算法容易并行, 硬件计算复杂度很低, 同时具有良好的压缩质量。目前, 应用该算法的数据压缩机工程样机已完成方案阶段的联试实验。比特平面扫描过程由于上下关联太强, 并行处理难度较大, 对它的数据结构的研究还在进行中。

参考文献:

- [1] Malvar H S. Biorthogonal and Nonuniform Lapped Transforms for Transform Coding with Reduce Blocking and Ringing Artifacts[J]. IEEE Transactions on Signal Processing, 1998, 46(4): 1043-1053.
- [2] Tran T D, Jie L, Tu C. Lapped Transform Via Time-domain Pre- and Post-filtering[J]. IEEE Transactions on Signal Processing, 2003, 51(6): 1557-1571.
- [3] Tran T D. The BinDCT: Fast Multiplierless Approximation of the DCT[J]. IEEE Signal Processing Letters, 2000, 7(6): 141-155.
- [4] 钟广军, 成礼智, 陈火旺. 双正交重叠变换的整数实现算法与图像压缩[J]. 电子学报, 2001, 29(11): 1475-1477.
- [5] Zhao D B, Gao W. Low Complexity and Low Memory Entropy Coder for Image Compression[J]. IEEE Trans. on Circuit and System for Video Technology, 2001, 11(10): 1140-1145.
- [6] Tu C, Tran T D. Context-based Entropy Coding of Lock Transform Coefficients for Image Compression[J]. IEEE Transactions on Image Processing, 2002, 11(11): 1271-1283.